НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)

"___" червня 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напряму підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: «КЛІЄНТ-СЕРВЕРНА СИСТЕМА МОНІТОРИНГУ ТА АНАЛІЗУ СТАНУ IP-МЕРЕЖІ»

Виконав : студент IV курсу, групи КВ-53
(шифр групи)

Булах Олександр Віталійович _____ _____
(прізвище, ім'я, по батькові) (підпис)

Керівник доц., к.т.н., доц. Щербина О.А. _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент професор кафедри ОТ, д.т.н., проф. Кулаков Ю.О. ___ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.050102 «Комп’ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)
«___» червня 2019 р.

ЗАВДАННЯ
на дипломний проект студента
Булаха Олександра Віталійовича

1. Тема проекту: Клієнт-серверна система моніторингу та аналізу стану IP - мережі.

Керівник проекту Щербина Олександр Андрійович, доц. каф. СПіСКС, к.т.н., затверджені наказом по університету від «22» травень 2019 р. №1330-С

2. Термін подання студентом проекту _____

3. Вихідні дані до проекту: див. технічне завдання.

4. Зміст пояснювальної записки: аналіз існуючих рішень та обґрунтування теми дипломного проекту, способи написання систем моніторингу пакетів даних комп’ютерної мережі та використання клієнт-серверної архітектури для реалізації програми, аналіз роботи програмного продукту.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо): схема взаємодій модулів системи, клієнт-

серверна система моніторингу мережі, процес роботи системи моніторингу, процес роботи клієнт-серверної системи.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, к.т.н.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
	Видача завдання на дипломне проектування	10.10.2018	
	Розробка технічного завдання	25.10.2018	
	Аналіз існуючих рішень	15.11.2018	
	Вибір середовища розробки	04.12.2018	
	Розробка програмного продукту	25.02.2019	
	Відлагодження програмного продукту	19.03.2019	
	Підготовка пояснювальної записки	29.04.2019	
	Оформлення матеріалів проекту	25.05.2019	
	Попередній огляд матеріалів диплому на кафедрі	30.05.2019	

Студент _____ _____
(підпис) (ініціали, прізвище)

Керівник проекту _____ _____
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Дипломний проект включає пояснювальну записку (50 стор., 15 рис.).

В бакалаврському проекті розроблено клієнт-серверну систему моніторингу та аналізу стану IP-мережі. Розроблена система дозволяє на практиці засвоїти знання з курсу «Комп'ютерних мереж».

Клієнтська частина системи являє собою сайт, який містить весь необхідний функціонал для зручного користування ресурсом. Серверна частина – забезпечує обробку запитів клієнта, формування ICMP повідомлень, для моніторингу стану будь-якої IP адреси, що знаходиться в сегменті IPv4, реалізована підтримка пошуку IP адрес через DNS сервери.

Система отримує відповідь на свої ICMP повідомлення, шукає їх з усіх прийнятих пакетів та виводить їх для аналізу. Тільки зареєстровані користувачі мають змогу перевіряти стан IP адрес, а гостьові користувачі, лише переглядати попередні запити.

У даному дипломному проекті розроблено: архітектуру клієнт-серверної системи на основі web-ресурсу в якому виконані, алгоритм авторизації користувача, процедура синхронізації даних, процедура моніторингу IP-адрес та дизайн web-сторінок.

Ключові слова:

IP, IPv4, Python, Django, ICMP, NMS, клієнт-серверна система, web-ресурс.

ABSTRACT

The diploma project includes an explanatory note (50 pages, 15 pictures).

The bachelor project has developed a client-server system for monitoring and analyzing the state of the IP network. The developed system allows to practice knowledge of the course "Computer Networks" in practice.

The client part of the system is a site that contains all the necessary functionality for easy use of the resource. The server part - provides processing of client requests, ICMP message generation, to monitor the status of any IP address located in the IPv4 segment, IP address search support is supported through DNS servers.

The system receives an answer to its ICMP message, searches for it from all accepted packets and displays them for analysis. Only registered users can check the status of IP addresses, and guest users only see previous queries.

This graduation project developed: the client-server system architecture based on the web-resource in which executed, user authentication algorithm, data synchronization procedure, IP-monitoring procedure and web-page design.

Keywords:

IP, IPv4, Python, Django, ICMP, NMS, client-server system, web-resource.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
1	А4	ІАЛЦ.467100.002 ТЗ	Клієнт-серверна	4		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Технічне завдання.			
2	А4	ІАЛЦ.467100.003 ТП	Клієнт-серверна	2		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Відомість технічного			
			проекту.			
3	А4	ІАЛЦ.467100.004 ПЗ	Клієнт-серверна	50		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Пояснювальна записка.			
4	А4	ІАЛЦ.467100.005 Е1	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Схема взаємодій модулів			
			системи. Структурна схема			

Змін.	Арк.	№ докум.	Підпис	Дата	**ІАЛЦ.467100.001 ОА**		
Розробив		Булах О.В.			«Клієнт-серверна система моніторингу та аналізу стану IP-мережі» **Опис альбому**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Щербина О.А.					
Н. контроль		Клятченко Я.М.				**КПІ ім. Ігоря Сікорського, ФПМ, КВ-53**	
Затвердив		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
5	А4	ІАЛЦ.467100.006 Е2	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Клієнт-серверна система			
			моніторингу мережі.			
			Схема структурна.			
6	А4	ІАЛЦ.467100.007 Д1	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Процес роботи системи			
			моніторингу.			
			Схема алгоритму.			
7	А4	ІАЛЦ.467100.008 Д2	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Процес роботи клієнт-			
			серверної системи.			
			Схема алгоритму.			
8		Диск CR-ROM	Текст пояснювальної	1		
			записки.			
			Графічні матеріали			

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ……2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ……2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ...……2
4. ДЖЕРЕЛА РОЗРОБКИ……..2
5. ТЕХНІЧНІ ВИМОГИ……2
   5.1 Вимоги до продукту, що розробляється...……2
   5.2 Вимоги до програмного забезпечення……..3
   5.3 Вимоги до апаратного забезпечення…….3
6. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ ……3
7. ЕТАПИ РОЗРОБКИ……...……..4

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування роботи – «Клієнт-серверна система моніторингу та аналізу стану IP-мережі».

Область застосування: використання в повсякденному житті для перевірки доступу до будь-якого ресурсу через сегмент мережі IPv4.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання дипломного проекту першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення клієнт-серверної системи для моніторингу та аналізу комп'ютерної мережі, для підтримки курсу «Комп'ютерних мереж» та для використання у повсякденному житті.

4. ДЖЕРЕЛА РОБОТИ

Джерелами роботи є конспект лекцій з курсу «Комп'ютерних мереж», науково-технічна література CISCO.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до продукту, що розробляється:

Клієнтська частина системи:

- реєстрація та вхід користувачів у систему;
- заповнити форму для запуску серверної частини, яка приймає IP адресу;
- виведення на екран інформацію про стан з'єднання з IP адресою, у вигляді отриманих ICMP повідомлень для подальшого аналізу.

Серверна частина системи:

- на вхід приймає або IP адресу, або Domain ім’я;
- у разі прийняття Domain ім’я перетворює його на IP адресу;
- обмінюється ICMP повідомленнями з сервером за даною адресою;
- з усіх отриманих і відправлених пакетів, шукає лише необхідні повідомлення;
- в разі виявлення помилки з боку сервера за даною адресою, повідомляє про це;
- на вихід передаються розібрані ICMP пакети, які можна проаналізувати.

5.2 Вимоги до програмного забезпечення:

- операційна система Linux;
- python3.7;
- django;
- браузер.

5.3 Вимоги до апаратного забезпечення:

- монітор;
- комп’ютер;
- мобільний пристрій.

6. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проекту повинна бути розроблена наступна документація:

- пояснювальна записка;
- керівництво користувача;
- креслення:
  - Організація блоків клієнтської частини. Схема структурна;
  - Організація блоків серверної частини. Схема структурна;
  - Принцип роботи системи моніторингу. Схема структурна;
  - Алгоритм пошуку IP адреси. Схема структурна.

7. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання
1.	Видача завдання на дипломне проектування	10.10.2018
2.	Розробка технічного завдання	25.10.2018
3.	Аналіз існуючих рішень	15.11.2018
4.	Вибір середовища розробки	04.12.2018
5.	Розробка програмного продукту	25.02.2019
6.	Відлагодження програмного продукту	19.03.2019
7.	Підготовка пояснювальної записки	29.04.2019
8.	Оформлення матеріалів проекту	25.05.2019
9.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
1	А4	ІАЛЦ.467100.004 ПЗ	Клієнт-серверна	50		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Пояснювальна записка			
2	А4	ІАЛЦ.467100.005 Е1	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Схема взаємодій модулів			
			системи. Структурна схема			
3	А4	ІАЛЦ.467100.006 Е2	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Клієнт-серверна система			
			моніторингу мережі.			
			Схема структурна.			

Змін.	Арк.	№ докум.	Підпис	Дата	**ІАЛЦ.467100.003 ТП**		
Розробив		Булах О.В.			«*Клієнт-серверна система моніторингу та аналізу стану IP-мережі*» ***Відомість технічного проекту***	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Щербина О.А.					
Н. контроль		Клятченко Я.М.				***КПІ ім. Ігоря Сікорського, ФПМ, КВ-53***	
Затвердив		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
4	А4	ІАЛЦ.467100.007 Д1	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Процес роботи системи			
			моніторингу.			
			Схема алгоритму.			
5	А4	ІАЛЦ.467100.008 Д2	Клієнт-серверна	1		
			система моніторингу			
			та аналізу стану			
			IP-мережі.			
			Процес роботи клієнт-			
			серверної системи.			
			Схема алгоритму.			
6		Диск CR-ROM	Текст пояснювальної	1		
			записки.			
			Графічні матеріали			

Змін.	Арк.	№ докум.	Підпис	Дата	***ІАЛЦ.467100.003 ТП***	Арк. 2

ЗМІСТ

стор.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ ……..…3

ВСТУП……..……4

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ …….…….5

1.1 Аналіз актуальності задачі ...…….…..……..5

1.1.1 Класифікація комп’ютерних мереж за територіальною розповсюдженістю ...……...…….……...5

1.1.2 Класифікація комп’ютерних мереж за архітектурою ……..…….7

1.2 Модель OSI ...…….……...12

1.2.1 Рівні моделі OSI ...…….…….……13

2. СТРУКТУРА МЕРЕЖЕВОГО РІВНЯ МОДЕЛІ OSI. АНАЛІЗ СПОСОБІВ МОНІТОРИНГУ З ВИКОРИСТАННЯМ МОВИ PYTHON ……...……...…….……....16

2.1 Мережевий рівень моделі OSI ...…….……16

2.1.1 Функції мережевого рівня ...…….…….……17

2.1.2 Інтернет-протокол (IP) …….…….……..…17

2.2 Сегмент комп’ютерної мережі IPv4 …….……...……23

2.3 Моніторинг мережі з використанням Python …….…….……32

2.3.1 Різні типи рішень моніторингу мережі ...…….….33

2.3.2 Загальні методи та протоколи моніторингу ...…….…...…....34

3. АНАЛІЗ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ ……37
3.1 Створення web-ресурсу за допомогою Python Django ……37
3.2 Створення програми моніторингу стану мережі ……45
3.3 Опис роботи і зв’язків ресурсу та програми моніторингу ...….……46
3.4 Приклад роботи. Демонстрування можливостей системи ...……..47
ВИСНОВКИ …….. ……..…….49
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ……50

ДОДАТКИ
Додаток 1. Копії графічних матеріалів
– ІАЛЦ.467100.005 Е1. Схема вза’ємодій модулів системи. Структурна схема;
– ІАЛЦ.467100.006 Е2. Клієнт-серверна система моніторингу мережі. Схема структурна.;
– ІАЛЦ.467100.007 Д1. Процес роботи системи моніторингу. Схема алгоритму;
– ІАЛЦ.467100.008 Д2. Процес роботи клієнт-серверної системи. Схема алгоритму.
Додаток 2. Лістинг програми
Додаток 3. Презентація проекту
Додаток 4. Довідка про впровадження

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

AS	Автономні системи
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
MIB	Інформаційна база управління
NMS	Система моніторингу мережі
OSI	Open System Interconnection
SNMP	Простий протокол керування мережею

ВСТУП

Останніми десятиліттями не можна уявити своє життя без комп'ютерних мереж. Комунікація є основною необхідністю людей, а комп'ютерна мережа робить це максимально просто й доступно, з будь-якої точки світу можна відправити повідомлення й воно буде доставлено за лічені мілісекунди, а іноді навіть й мікросекунди.

Комп'ютерна мережа – це система, що забезпечує обмін даними між обчислювальними пристроями (комп'ютерами, серверами, маршрутизаторами та іншим обладнанням). Для передачі інформації можуть використовуватись різні середовища передачі даних.

Internet Protocol (IP) став тим протоколом, що об'єднав окремі комп'ютерні мережі в глобальну мережу Інтернет. Є декілька версій протоколу такі, як IPv4 та IPv6, що розбивають Інтернет, на окремі сегменти, що мають різні правила фрагментації та оформлення пакету даних. Протокол IPv4 більш розповсюджений, оскільки був вперше описаний 1981 року й став першою широко використовуваною версією IP.

Моніторинг комп'ютерної мережі означає постійне спостереження за нею та пошуку помилок передачі даних, або занадто повільних сегментів, що спричиняють занадто великий час очікування відповіді.

На сьогоднішній день для моніторингу мережі використовують простий протокол керування мережею SNMP(Simple Network Management Protocol). Ця технологія створена для спрощення процесу адміністрування підмережі. Вручну і індивідуально входити в сотні або тисячі вузлів буде надзвичайно трудомістким і ресурсномістким. Для порівняння, використання SNMP з NMS(Network Monitoring System) дозволяє адміністратору мережі керувати та контролювати всі ці вузли з єдиної станції.

Метою роботи є створення системи, за допомогою якої будь-який користувач зможе дізнатись причину повільної передачі повідомлень.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Аналіз актуальності задачі

Комп’ютерна мережа – це ряд з’єднань і пов’язаних з ними пристроїв, через які комп’ютери можуть обмінюватись інформацією один з одним. Комп’ютерна мережа складається з двох або більше комп’ютерів, які з’єднані між собою для спільного використання ресурсів, обміму файлами або дозволу електронних комунікацій. У комп’ютерній мережі окремі станції мають назву – вузли. Вони можуть бути комп’ютерами, терміналами, або блоками зв’язку різних видів. Комп’ютери в мережі можуть бути з’єднаними через кабелі, телефонні лінії, радіоканали, супутникові, або інфрачервоні світлові промені. На додаток до фізичного підключення вузлів, мережа має функцію створення згуртованої архітектури, що дозволяє практично безперервно передавати дані при використанні різних типів обладнання. Open System Interconnection (OSI) та IBM’s System Network Architecture - це дві популярні архітектури, що використовуються в даний час.

Оскільки у наш час, неможливо уявити своє життя без використання комп’ютерних мереж, нам необхідний зручний спосіб, для визначення якості з’єднання з тим чи іншим ресурсом.

Розглянемо наступні класифікації комп’ютерних мереж, для більшого розуміння їх застосування.

1.1.1 Класифікація комп’ютерних мереж за територіальною розповсюдженістю

1) BAN (Body Area Network) – мережа комп’ютерних пристроїв розташованих на тілі носія, або імплантованих у нього.

2) PAN (Personal Area Network) – персональна мережа, необхідна для взаємодії різноманітних пристроїв одного користувача.

3) LAN (Local Area Network) – локальна мережа, що має замкнуту інфраструктуру. Вони є мережами закритого типу, доступ до яких дозволено лише обмеженій кількості користувачів, для яких робота в цій мережі безпосередньо пов’язана з їх професійною діяльністю.[3]

4) CAN (Campus Area Network) – кампусна мережа, об’єднання локальних мереж сусідніх будівель.

5) MAN (Metropolitan Area Network) – міська мережа, у межах одного або кількох сусідніх міст, об’єднання багатьох локальних обчислювальних мереж.[3]

6) WAN (Wide Area Network) – глобальна мережа, що покриває великі географічні регіони, складається як з локальних мереж, так й з інших телекомунікаційних мереж та пристроїв. Глобальні мережі є відкритими та орієнтовані на обслуговування будь-яких користувачів. [3]

Основними типами мережі є локальна (LAN – Local Area Network) та глобальна (WAN – Wide Area Network).

Локальна мережа – це комп’ютерна мережа, яка охоплює локальну область. Це може буди будинок, офіс або невелика група будівель, тощо. Топологія мережі диктує її фізичну структуру. Загальноприйнятий максимальний розмір для локальної мережі складає 1 км^2. В даний час існують дві поширені технології підключення для локальної мережі – Ethernet та Token Ring. Локальна мережа зазвичай складається з двох або більше комп’ютерів, принтерів, пристроїв зберігання великої ємкості, які називають файловими серверами, які забезпечують доступ кожного комп’ютера в мережі до загального набору файлів. Даний тип мережі керується програмним забезпеченням локальної мережі. Користувачі LAN можуть також мати доступ до інших локальних мереж, або підключатися до глобальної мережі. Локальні мережі з подібними архітектурами пов’язані між собою точками передачі, які називаються «мостами», а локальні мережі з різними архітектурами використовують «шлюзи» для перетворення даних

під час проходження між системами. Маршрутизатори використовуються для встановлення з’єднання між локальними мережами[4].

Глобальна мережа (WAN) – це комп’ютерна мережа, що охоплює широку географічну зону і включає велику кількість комп’ютерів. Комп'ютерні мережі можуть з'єднувати комп'ютери за допомогою кабелів, оптичних волокон або супутників і модемів.

Як правило, глобальні мережі – це сукупність декількох локальних мереж та/або міських мереж. Це означає, що користувач з однієї локальної мережі може обмінюватися інформацією з користувачем, що з’єднаний з іншою LAN, якщо ці дві мережі утворюють, або підключені до глобальної мережі.

Найкращим прикладом глобальної мережі є Інтернет, сукупність мереж і шлюзів, що зв'язують мільйони користувачів комп'ютерів на кожному континенті. Мережі в Інтернет пов'язані спільними комунікаційними програмами та протоколами. Протокол - це набір встановлених стандартів, які дозволяють комп'ютерам спілкуватися один з одним. Для мереж WAN можна використовувати ряд мережевих протоколів, таких як TCP / IP, X.25, ATM і Frame relay, тощо. За допомогою Інтернету користувачі можуть отримувати різноманітну інформацію за допомогою посилань, виділеного тексту або складного програмного забезпечення для пошуку, відомого як пошукові системи.[4]

1.1.2 Класифікація комп’ютерних мереж за архітектурою

1) Клієнт-сервер;
2) Однорангова мережа.

Клієнт- сервер – це обчислювальна або мережева архітектура, в якій завдання, чи навантаження розподіленні між постачальниками послуг, названими серверами та замовниками послуг, названими клієнтами. Фактично клієнт та сервер – це програмне забезпечення. Зазвичай такі програми розташовані на різних обчислювальних машинах та взаємодіють

через комп’ютерну мережу за допомогою мережевих протоколів, але це не означає, що вони не можуть бути розташованими на одній машині. Програми-сервери очікують від клієнтських програм запити й надають їм свої ресурси у вигляді даних, чи у вигляді сервісних функцій. Оскільки одна програма-сервер може виконувати запити від множини програм-клієнтів, її розміщують на окремій робочій станції, налаштованій спеціальним чином, як правило, сумісно з іншими програмами-серверами, тому швидкодія даної машини повинна бути досить високою.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися спочатку до одного сервера, а потім до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Архітектура клієнт / сервер - це обчислювальна модель, в якій сервер розміщує, доставляє і управляє більшістю ресурсів і послуг, які використовуються клієнтом для досягнення його цілей. Цей тип архітектури має один або більше клієнтських комп'ютерів, підключених до центрального сервера через мережу або підключення до Інтернету. Ця система розподіляє обчислювальні ресурси.

Архітектура клієнт / сервер також відома як мережева обчислювальна модель або мережа клієнт / сервер, оскільки всі запити та послуги доставляються через мережу.

Архітектура клієнт / сервер - це архітектура виробників / споживачів, де сервер виступає як виробник і клієнт як споживач. Сервер розміщує і надає клієнтові послуги високого класу на вимогу. Ці послуги можуть включати доступ, зберігання, спільний доступ до файлів, доступ до принтера та / або прямий доступ до необробленої обчислювальної потужності сервера.

Архітектура клієнт / сервер працює, коли клієнтський комп'ютер посилає запит ресурсу або процесу на сервер через мережеве з'єднання, яке потім обробляється і доставляється клієнту. Серверний комп'ютер може керувати декількома клієнтами одночасно, тоді як один клієнт може бути підключений до декількох серверів одночасно, кожен з яких надає інший набір послуг. У найпростішій формі Інтернет також заснований на архітектурі клієнт / сервер, де веб-сервери обслуговують багатьох одночасних користувачів з даними веб-сайту.

Архітектура клієнт-сервер - це централізована система ресурсів, де сервер зберігає всі ресурси. Сервер отримує численні виступи на своєму краю для спільного використання ресурсів своїм клієнтам на запит. Клієнт і сервер можуть бути на одній або в мережі. Сервер глибоко стабільний і масштабований, щоб повертати клієнтам відповіді. Ця архітектура орієнтована на послуги, що означає, що обслуговування клієнтів не буде перервано. Архітектура клієнт-сервер підпорядковує мережевому трафіку, реагуючи на запити клієнтів, а не на повну передачу файлів. Він відновлює файловий сервер сервером баз даних.

Клієнтські комп'ютери реалізують зв'язок, щоб дозволити користувачеві комп'ютера запитувати послуги сервера і представляти результати, які повертає сервер. Сервери чекають появи запитів від клієнтів і повертають їх. Сервер, як правило, надає клієнтам стандартизований простий інтерфейс, щоб уникнути плутанини апаратного та програмного забезпечення. Клієнти розташовані на робочих місцях або на особистих машинах, в той же час сервери будуть розташовані десь потужні в мережі. Ця архітектура корисна в основному, коли клієнти та сервер мають окремі завдання, які вони виконують. Багато клієнтів можуть одночасно отримувати інформацію про сервер, а також клієнтський комп'ютер може виконувати інші завдання, наприклад, надсилати електронні листи.

Типи архітектури клієнт-сервера

1) 1-ярусна архітектура

У цій категорії налаштувань клієнт-сервер користувальницький інтерфейс, маркетингова логіка і логіка даних присутні в одній системі. Такий сервіс є розумним, але їм важко керувати через відхилення даних, які виділяють реплікацію роботи.

Наприклад, презентація, бізнес, рівні доступу до даних в одному пакеті програм. Ці дані зазвичай зберігаються в локальній системі або на спільному диску. Програми, які обробляють всі три яруси, такі як MP3-плеєр, MS Office, підпадають під однорівневу програму.

2) Дворівнева архітектура

У цьому типі клієнт-серверного середовища інтерфейс користувача зберігається на клієнтській машині, і база даних зберігається на сервері. Логіка бази даних та бізнес-логіка зберігаються на будь-якому клієнті або сервері, але їх потрібно підтримувати. Якщо бізнес-логіка та логіка даних збираються на стороні клієнта, вона називається архітектурою тонкого сервера товстих клієнтів. Якщо бізнес-логіку та логіку даних обробляються на сервері, це називається архітектурою товстого сервера тонкого клієнта. Це вважається доступним.

У дворівневій архітектурі клієнт і сервер повинні входити в безпосереднє об'єднання. Якщо клієнт вводить дані на сервер, не повинно бути жодного проміжного. Це робиться для швидких результатів і уникнення плутанини між різними клієнтами. Наприклад, програмне забезпечення онлайн-бронювання квитків використовує цю дворівневу архітектуру (рисунок 1.1).

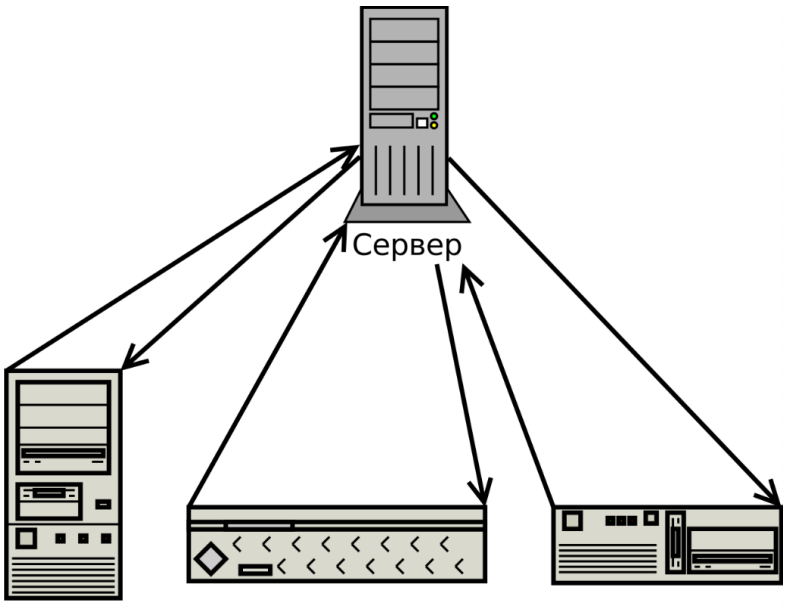

Рисунок 1.1 – Приклад дворівневої архітектури клієнт-сервер.

3) Трирівнева архітектура

У цьому різноманітному контексті клієнт-сервер використовується додаткове проміжне програмне забезпечення, що означає, що запит клієнта надходить на сервер через цей середній рівень, і відповідь сервера отримується першим, а потім клієнтом. Ця архітектура захищає 2-ярусну архітектуру і дає кращу продуктивність. Ця система є дорогою, але вона проста у використанні. Проміжне програмне забезпечення зберігає всю логіку бізнес-логіки та проходження даних. Ідея проміжного програмного забезпечення полягає в постановці баз даних, чергуванні, виконанні додатків, планування і т.д.

Трирівнева архітектура розбивається на 3 частини, а саме: рівень презентації (Client Tier), рівень прикладних програм (Business Tier) і рівень баз даних (Data Tier). Система клієнта керує рівнем презентації; Сервер додатків піклується про рівень програми, а серверна система контролює рівень бази даних.

У нинішньому сценарії онлайн-бізнесу зростають вимоги до швидкого реагування та якісних послуг. Тому комплексна клієнтська архітектура має вирішальне значення для ділової діяльності. Компанії, як правило, досліджують можливості для підтримки обслуговування та якості

для підтримки свого ринку за допомогою архітектури клієнт-сервер. Архітектура підвищує продуктивність завдяки практиці рентабельних користувальницьких інтерфейсів, поліпшенню зберігання даних, розширеному підключенню і захищеним сервісам.

1.2 Модель OSI

Модель OSI (Open Systems Interconnection) - концептуальна модель, яка характеризує і стандартизує функції зв'язку телекомунікаційної або обчислювальної системи без урахування її внутрішньої структури та технології. Її метою є сумісність різноманітних систем зв'язку зі стандартними протоколами. Це правильне визначення моделі OSI. Визначення виглядає досить простим і доречним, але коли справа доходить до роз'яснення моделі OSI, багато людей заплутуються і не можуть зрозуміти, що це таке. Перше, що приходить на думку людей, коли вони чують модель OSI, це те, що їм доведеться багато запам'ятовувати і зупинятися. Це було саме те, що відбулося, коли кожен вперше прочитав про рівні OSI.[5]

OSI означає "Open Systems Interconnection". Точне визначення моделі OSI вже наведено вище. Якщо говорити простіше, модель OSI є інструментом, який використовують IT-фахівці для моделювання або відстеження фактичного потоку передачі даних у мережах. Таким чином, модель OSI є логічною моделлю / уявленням про те, як мережеві системи повинні передавати дані (або спілкуватися) один з одним.

Модель OSI розбиває цю процедуру передачі / обміну даними на різні компоненти (так звані рівні). Чому рівні, тому що ці компоненти слідують належному порядку виконання. Наприклад, фізичний рівень, в якому відбуваються «фізичні» підключення та підключення, рівень передачі даних, в якому відбувається перемикання, і т.д. В цілому, існує сім рівнів, які разом складають модель OSI.[5]

Метою еталонної моделі OSI є керівництво над постачальниками та розробниками таким чином, щоб продукти цифрового зв'язку та програмні продукти, які вони створюють, з якими взаємодіють відповідали чітким нормам засобів комунікації. Більшість постачальників, які беруть участь у телекомунікаціях, намагаються описати свої продукти та послуги стосовно моделі OSI, оскільки вона є еталонною.

Крім того, критично важливо, щоб IT-професіонал мав чітке уявлення про модель OSI. Це пов'язано з тим, що у випадку деяких мережевих проблем, використовуючи рівні OSI, вони можуть звузитися і з'ясувати, в якій частині проблема. Отже, використання підходу OSI Layered для усунення проблем мережі є дуже корисним.

Переваги моделі OSI

Це створює спільну платформу для розробників програмного забезпечення та виробників апаратних засобів, які заохочують створення мережевих продуктів, які можуть спілкуватися один з одним через мережу.

Вона допомагає адміністраторам мережі розділяти процес обміну великими даними в менших сегментах.

Завдяки незалежності рівнів, вона запобігає впливу змін на одиному рівні на інші.

Стандартизація мережевих компонентів дуже добре структурує функції, характерні для кожного рівня. Це знижує складність і прискорює еволюцію. Це спрощує викладання та навчання.

Як вже обговорювалося в розділі, модель OSI складається з семи рівнів, починаючи прикладним рівнем, який є найбільш близьким до кінцевого користувача, у верхній частині та закінчується фізичним рівнем, де відбувається фактична передача даних з використання середовища передачі. Розглянемо всі рівні окремо.

1.2.1 Рівні моделі OSI.

7. Рівень прикладних сервісів.

Це верхній рівень у семи рівнях OSI. Це рівень, з яким фактично взаємодіє кінцевий користувач (може бути програміст або звичайний користувач ПК). Цей рівень дозволяє отримати доступ до мережевих ресурсів.

6. Рівень представлення даних.

Це рівень, в якому операційна система працює з даними. Основними функціями цього рівня є переклад, шифрування та стиснення даних. В основному користувач взаємодіє з прикладним рівнем, який передає дані до рівня презентації.

5. Рівень сеансу

Цей рівень має завдання підтримувати належну комунікацію шляхом встановлення, керування та завершення сеансів між двома комп'ютерами. Наприклад, коли відвідуємо будь-який веб-сайт, наш комп'ютер повинен створити сеанс із веб-сервером цього веб-сайту.

4. Транспортний рівень

Цей рівень має дуже важливі функції. Він вирішує, скільки інформації потрібно надсилати одночасно. Таким чином, коли спілкуєтеся з веб-сайтом, цей рівень вирішить, скільки даних можете передати та отримати в певний момент часу. Крім того, цей рівень забезпечує надійний процес для обробки доставки повідомлень і відновлення помилок.

3. Мережевий рівень

Основна робота цього рівня полягає в переміщенні пакетів від джерела до пункту призначення і забезпечення міжмережевого зв'язку. Це рівень, на якому працюють маршрутизатори. Оскільки маршрутизатори працюють на мережевому рівні, то можна сказати, що IP-адреса знаходиться на рівні мережі.

2. Рівень передачі даних

Цей рівень відповідає за організацію бітів у кадри та забезпечення хоп-доставки. Це рівень, на якому працюють комутатори. Оскільки маршрутизатори працюють на мережевому рівні, можемо сказати, що MAC-адреса знаходиться на рівні каналу передачі даних. Всі комп'ютери в певній мережі підключаються до перемикача inito, щоб вони могли спілкуватися один з одним.

1. Фізичний рівень

Це рівень, на якому відбувається реальна передача бітів даних через середовище. Цей рівень, як випливає з назви, містить усі фізичні речі, які з'єднують комп'ютери разом.

Оскільки для моніторингу стану зв’язку між вузлами системи необхідно аналізувати структуру та зміст пакетів відправлених за певною IP адресою. Тому для цього необхідно детально розглянути організацію протоколів мережевого рівня, а саме для дипломного проекту – протокол IPv4.

2.СТРУКТУРА МЕРЕЖЕВОГО РІВНЯ МОДЕЛІ OSI. АНАЛІЗ СПОСОБІВ МОНІТОРИНГУ З ВИКОРИСТАННЯМ МОВИ PYTHON

2.1 Мережевий рівень моделі OSI

Мережний рівень - це третій рівень моделі взаємодії відкритих систем (OSI Model) і рівень, який забезпечує маршрутизацію даних. Дані передаються у вигляді пакетів через логічні мережеві контури в упорядкованому форматі, контрольованому мережевим рівнем.

Налаштування логічного з'єднання, перенаправлення даних, маршрутизація та повідомлення про помилки доставки - це основні функції мережевого рівня.

Мережевий рівень вважається основою моделі OSI. Він вибирає і управляє найкращим логічним шляхом для передачі даних між вузлами. Цей рівень містить апаратні пристрої, такі як маршрутизатори, мости, брандмауери та комутатори. Він фактично створює логічні з’єднання найбільш ефективних маршрутів для передачі даних та реалізує їх через фізичні носії.

Протоколи мережного рівня існують у кожному хості або маршрутизаторі. Маршрутизатор перевіряє поля заголовків всіх IP-пакетів, які проходять через нього.

Інтернет-протокол і Netware IPX / SPX є найбільш поширеними протоколами, пов'язаними з мережним рівнем.

У моделі OSI мережевий рівень реагує на запити з рівню над ним (транспортний рівень) і видає запити рівню, розташованому під ним (рівень каналу передачі даних).

Розташована на рівні 3 моделі взаємодії відкритих систем (OSI), основною функцією мережевого рівня є переміщення даних у інші мережі та через них. Протоколи мережевого рівня досягають цієї мети шляхом упаковки даних з коректною інформацією про мережеві адреси, вибору відповідних мережевих маршрутів і пересилання упакованих даних по стеку до транспортного рівня (Layer 4). Існуючі протоколи, які взагалі

відображають мережевий рівень OSI, включають частину IP протоколу управління передачею / Інтернет-протокол (TCP / IP) - як IPv4, так і IPv6 - а також мережевий пакетний обмін NetWare / обмін пакунками з послідовною передачею SPX).

Інформація про маршрут, що міститься в пакеті, включає в себе джерело хоста-відправника і кінцеве призначення віддаленого вузла. Ця інформація міститься в заголовку мережевого рівня, який інкапсулює мережеві кадри на рівні каналу передачі даних (Layer 2). Ключова відмінність - і важливість - між транспортною інформацією, що міститься на рівні 2, у порівнянні з транспортною інформацією, що міститься в мережевому рівні, полягає в тому, що інформація може виходити за межі локальної мережі для досягнення хостів у віддалених мережевих місцях.

2.1.1 Функції мережевого рівня

Основна функція мережевого рівня полягає в тому, щоб дозволити взаємодію різних мереж. Це здійснюється шляхом пересилання пакетів на мережеві маршрутизатори, які спираються на алгоритми для визначення найкращих шляхів для переміщення даних. Ці шляхи відомі як віртуальні схеми. Мережевий рівень покладається на протокол ICMP (Internet Control Message Protocol) для обробки та діагностики помилок для забезпечення правильного надсилання пакетів. Якість обслуговування (QoS) також доступна для того, щоб визначити пріоритет певного трафіку над іншим трафіком. Мережевий рівень може підтримувати або орієнтовані на з'єднання, або мережі без з'єднання, але така мережа може бути тільки одного типу, а не обох.

2.1.2 Інтернет-протокол (IP)

У мережах протокол є стандартизованим способом виконання певних дій і даних форматування, так що два чи більше пристроїв можуть спілкуватися і розуміти один одного.

Інтернет-протокол (IP) - це основний набір (або протокол зв'язку) форматів цифрових повідомлень і правил для обміну повідомленнями між комп'ютерами через одну мережу або ряд взаємопов'язаних мереж, використовуючи пакет Інтернет-протоколу (який часто називають TCP / IP) . Повідомлення обмінюються як дейтаграми, також відомі як пакети даних або просто пакети.

IP є первинним протоколом в Інтернет-рівні Internet Protocol Suite, який являє собою набір протоколів зв'язку, що складаються з чотирьох шарів абстракції: шар зв'язку (найнижчий), Інтернет-шар, транспортний шар і прикладний рівень (найвищий).

Основною метою і завданням IP є доставка дейтаграм з вихідного хоста (вихідного комп'ютера) до хоста призначення (на приймальному комп'ютері) на основі їх адрес. Для досягнення цієї мети IP включає в себе методи і структури для розміщення тегів (адресну інформацію, яка є частиною метаданих) у дейтаграм. Процес введення цих тегів у дейтаграми називається інкапсуляцією.

Подумайте про анологію з поштовою системою. IP схожий на американську поштову систему тим, що дозволяє адресовувати пакет (дейтаграмму) (інкапсуляцію) і вводити в систему (Інтернет) відправник (вихідний хост). Однак немає прямого зв'язку між відправником і одержувачем.

Пакет (дейтаграма) майже завжди розділяється на частини, але кожен фрагмент містить адресу одержувача (хоста призначення). Зрештою, кожен шматок прибуває до приймача, часто різними маршрутами і в різний час. Ці маршрути і час також визначаються поштовою системою, яка є ІС. Однак поштова система (в транспортному і прикладному рівнях) з'єднує всі частини перед доставкою одержувачу (адресату).

Примітка: IP - це фактично протокол без з'єднання, що означає, що маршрут до приймача (хоста призначення) не потрібно встановлювати перед передачею (вихідним хостом).

Спочатку IP-служба була безсистемною службою дейтаграм в програмі керування передачею, створеною Vint Cerf і Bob Kahn в 1974 році. Вони разом створили Internet Protocol Suite, який часто називають TCP / IP.

Інтернет-протокол версії 4 (IPv4) був першою основною версією IP. Це домінуючий протокол Інтернету. Однак, iPv6 є активним і використовується, і його розгортання зростає в усьому світі.

Адреси та маршрутизація є найбільш складними аспектами IP. Проте інтелект в мережі розташований на вузлах (точках мережевого з'єднання) у вигляді маршрутизаторів, які пересилають дейтаграми на наступний відомий шлюз на маршруті до кінцевого пункту призначення. Маршрутизатори використовують протоколи внутрішнього шлюзу (IGP) або протоколи зовнішнього шлюзу (EGP), що допомагають у прийнятті рішень про маршрутизації. Маршрути визначаються префіксом маршрутизації в дейтаграмах. Таким чином, процес маршрутизації може стати складним. Але з неймовірною швидкістю маршрутизація визначає найкращий маршрут, а частини дейтаграми і дейтаграми в кінцевому підсумку прибувають до місця призначення.

Інтернет-протокол (IP) - це метод або протокол, за допомогою якого дані передаються з одного комп'ютера на інший через Інтернет. Кожен комп'ютер (відомий як хост) в Інтернеті має принаймні одну IP-адресу, яка унікально ідентифікує його з усіх інших комп'ютерів в Інтернеті.

Коли Ви надсилаєте або отримуєте дані (наприклад, повідомлення електронної пошти або веб-сторінку), повідомлення поділяється на невеликі порції, які називаються пакетами. Кожен з цих пакетів містить як Інтернет-адресу відправника, так і адресу одержувача. Будь-який пакет передається спочатку до комп'ютера шлюзу, який розуміє невелику частину Інтернету. Комп'ютер шлюзу зчитує адресу призначення і пересилає пакет до сусіднього шлюзу, який, в свою чергу, читає адресу призначення і так далі через Інтернет, поки один шлюз не розпізнає пакет як приналежність до

комп'ютера в його безпосередньому сусідстві або домені. Потім цей шлюз пересилає пакет безпосередньо на комп'ютер, адреса якого вказана.

Оскільки повідомлення поділяється на декілька пакетів, кожен пакет може, при необхідності, бути відправлений іншим шляхом через Інтернет. Пакети можуть надходити в іншому порядку, ніж порядок, в якому вони були відправлені. Інтернет-протокол доставляє їх. Це залежить від іншого протоколу, протоколу управління передачею (TCP), щоб повернути їх у правильному порядку.

Як тільки пакети прибувають до місця призначення, вони обробляються по-різному в залежності від того, який транспортний протокол використовується в поєднанні з IP. Найбільш поширеними транспортними протоколами є TCP і UDP.

IP - це протокол без з'єднання, що означає, що між кінцевими точками, які спілкуються, немає постійного зв'язку. Кожен пакет, який подорожує через Інтернет, розглядається як незалежна одиниця даних без будь-якого відношення до будь-якої іншої одиниці даних. (Причина того, що пакети потрапляють у правильний порядок, полягає в TCP, орієнтованому на з'єднання протоколі, який відстежує послідовність пакетів у повідомленні.) У комунікаційній моделі відкритих систем (OSI), IP знаходиться в рівні 3 - мережевий рівень.

Найпоширенішою версією IP сьогодні є Internet Protocol Version 4 (IPv4). Однак IP-версія 6 (IPv6) також починає підтримуватися. IPv6 забезпечує набагато більшу кількість адрес і, отже, можливість набагато більше користувачів Інтернету. IPv6 включає можливості IPv4 і будь-який сервер, який може підтримувати пакети IPv6, також може підтримувати пакети IPv4.

Щоб зрозуміти, чому необхідні протоколи, розглянемо процес надсилання листа. На конверті адреси записуються в такому порядку: ім'я, адреса, місто, штат і поштовий індекс. Якщо конверт впав у поштову скриньку з першим поштовим індексом, а потім за адресою, за якою слідує

держава, то пошта не доставить її. Існує узгоджений протокол для написання адрес, щоб поштова система працювала. Так само всі пакети даних IP повинні подавати певну інформацію в певному порядку, і всі IP-адреси слідують за стандартизованим форматом.

IP-адреса - це унікальний ідентифікатор, призначений пристрою або домену, який підключається до Інтернету. Кожна IP-адреса - це ряд символів, наприклад "192.168.1.1". Через DNS-перетворювачі, які переводять імена, що читаються людиною, в IP-адреси, користувачі можуть отримувати доступ до веб-сайтів, не запам'ятовуючи цю складну серію символів. Кожен IP-пакет буде містити як IP-адресу пристрою або домену, що посилає пакет, так і IP-адресу призначеного одержувача, подібно до того, як і адреса призначення, і зворотний адресу включені в поштову скриньку.

Різниця протоколів IPv4 та IPv6

Четверта версія IP (коротко IPv4) була введена в 1983 році. Проте, як з автомобільними номерами, існує лише скінченна кількість можливих перестановок для формування номерів, тому постачання доступних адрес IPv4 вичерпано. Адреси IPv6 мають набагато більше символів і, отже, більше перестановок; однак IPv6 ще не повністю прийнятий, і більшість доменів і пристроїв все ще мають адреси IPv4.

IP-пакети створюються шляхом додавання IP-заголовка до кожного пакета даних, перш ніж він буде відправлений на своєму шляху. Заголовок IP - це лише ряд бітів (одиниць і нулів), і він записує кілька фрагментів інформації про пакет, включаючи IP-адресу відправлення і отримання. Заголовки IP також повідомляють:

- Довжина заголовка
- Довжина пакета
- Time To Live (TTL), або кількість мережевих стрибків, які пакет може зробити, перш ніж він буде відкинутий
- Який транспортний протокол використовується (TCP, UDP тощо)

- У загальній кількості є 14 полів для інформації в заголовках IPv4, хоча один з них є необов'язковим.

Інтернет складається з взаємопов'язаних великих мереж, кожен з яких відповідає за певні блоки IP-адрес; ці великі мережі відомі як автономні системи (AS). Різноманітні протоколи маршрутизації, включаючи BGP, допомагають маршрутизувати пакети через AS на основі їх IP-адрес призначення. Маршрутизатори мають таблиці маршрутизації, які вказують, яким АС пакети повинні проходити, щоб досягти бажаного місця призначення якомога швидше. Пакети подорожують від AS до AS, поки вони не досягнуть того, що відповідає за цільову IP-адресу. Це AS потім внутрішньо маршрутизує пакети до місця призначення.

Пакети можуть приймати різні маршрути в одному місці, якщо це необхідно, так само, як група людей, що їдуть до узгодженого пункту призначення, може приїхати різними дорогами.

Протокол управління передачею (TCP) є транспортним протоколом, що означає, що він диктує спосіб передачі та отримання даних. Заголовок TCP входить до частини даних кожного пакета, який використовує TCP / IP. Перед передачею даних TCP відкриває з'єднання з одержувачем. TCP гарантує, що всі пакети прибудуть в порядку, як тільки почнеться передача. Через TCP одержувач підтверджує отримання кожного пакета, що надходить. Відсутні пакети будуть надіслані знову, якщо отримання не буде підтверджено.

TCP призначений для надійності, а не швидкості. Оскільки TCP повинен переконатися, що всі пакети надходять у порядок, завантаження даних через TCP / IP може тривати довше, якщо деякі пакети відсутні.

TCP і IP були спочатку розроблені для спільного використання, і їх часто називають пакетом TCP / IP. Однак інші протоколи транспортування можуть використовуватися з IP.

Протокол користувацьких дейтаграм або UDP є іншим широко використовуваним транспортним протоколом. Це швидше, ніж TCP, але він

менш надійний. UDP не переконується, що всі пакети доставляються і в порядку, і не встановлює з'єднання перед початком або отриманням передач.

UDP / IP зазвичай використовується для потокового аудіо або відео, оскільки це випадки, коли ризик випадання пакетів (тобто відсутні дані) переважає потреба у збереженні передачі в реальному часі. Наприклад, коли користувачі переглядають відео в Інтернеті, не кожен піксель повинен бути присутнім для кожного кадру відео. Користувачі віддають перевагу відтворенню відео на звичайній швидкості, ніж сидять і чекають, коли кожен біт даних буде доставлено.

2.2 Сегмент комп'ютерної мережі IPv4

IPv4 - це протокол без з'єднання, що використовується для мереж з комутацією пакетів. Вона функціонує на основі моделі з найкращими зусиллями, в якій не гарантується ні поставка, ні забезпечення відповідного секвенування або уникнення дублювання доставки. Інтернет-протокол Версії 4 (IPv4) є четвертим переглядом Інтернет-протоколу і широко використовуваним протоколом для передачі даних через різні види мереж. IPv4 є протоколом без з'єднання, що використовується в мережах з комутацією пакетів, таких як Ethernet. Він забезпечує логічне з'єднання між мережевими пристроями, забезпечуючи ідентифікацію для кожного пристрою. Існує багато способів налаштувати IPv4 з усіма типами пристроїв, включаючи ручні та автоматичні конфігурації, залежно від типу мережі. IPv4 визначено і зазначено в публікації IETF RFC 791.

Інтернет-протокол версії 4 (IPv4) реалізує дві основні функції мережевого трафіку: 1) адресація пакетів і 2) фрагментація пакетів.

1) Маршрутизація дейтаграм

Маршрутизація - це процес вибору кращого шляху / шляхів у мережі, на якому ефективно надсилати дейтаграми IPv4.

2) Фрагментація і повторна збірка дейтаграм. Щоб зрозуміти, що таке фрагментація дейтаграм IPv4, спочатку слід знати термін MTU (Maximum Transmission Unit). MTU - це розмір (у байтах) найбільшого пакета або кадру, який може проходити через певний пристрій або карту NIC(Network Interface Card/Controller). Під час подорожі по мережі, щоб дістатися до місця призначення, дейтаграми Інтернет-протоколу версії 4 (IPv4) можуть знадобитися для обходу різних мереж з неоднорідними MTU. Коли дейтаграма є більшою, ніж MTU мережі, яку вона повинна проходити, вона розділена на менші фрагменти і надсилається окремо. На цільовому комп'ютері реконструюється фрагментована датаграма IPv4, і цей процес називається повторною збіркою.

IPv4 використовує 32-розрядні адреси для зв'язку Ethernet у п'яти класах: A, B, C, D і E. Класи A, B і C мають різну довжину біта для адресації мережевого хоста. Адреси класу D використовуються для багатоадресної розсилки, що дуже корисно для військових цілей, тоді як адреси класу E зарезервовані для майбутнього використання.

IPv4 використовує 32-бітну (4 байт) адресацію, що дає 2^{32} адреси. Адреси IPv4 записуються в точково-десятковому форматі, який складається з чотирьох октетів адреси, виражених окремо в десятковій формі і розділених періодами, наприклад, 192.168.1.5.

IP-адреси довжиною 32 біти, вони діляться на чотири октети (8 біт). Основне розуміння двійкової нумерація дуже корисне, якщо ви збираєтеся керувати IP-адресами в мережі, тому що зміни у значеннях з 32 бітів вказують або іншу IP-адресу мережі або IP-адресу хоста.

IP-адреса хоста визначає пристрій, до якого можуть надсилатися IP-пакети. IP-адреса мережі визначає конкретний сегмент мережі, до якого можна підключити один або більше хостів. Нижче наведено характеристики IP-адреси:

• IP-адреси довжиною 32 біти;

• IP-адреси розділені на чотири розділи по один байт (октет);

• IP-адреси, як правило, записуються у форматі, відомому як десяткове число.

Щоб забезпечити певну структуру способу присвоєння IP -адрес, IP-адреси розподіляються на класи. Кожен клас має діапазон IP-адрес. Діапазон IP-адрес у кожному класі визначається номером бітів, виділених мережевому розділу 32-бітової IP-адреси.

У таблиці 2.1 перелічено діапазони IP-адрес за класом і маски, пов'язані з кожним класом. Цифри з жирним шрифтом вказується мережевий розділ IP-адреси для кожного класу. Решта цифр доступні для хоста IP-адреси. Наприклад, IP-адреса 10.90.45.1 з маскою 255.0.0.0 розбивається на мережевий IP адреса 10.0.0.0 і IP-адреса хоста 0.90.45.1.

Таблиця 2.1 – Перелік діапазонів IP адрес за класом

Class	**Range**
A (range/mask in dotted decimal)	**0** .0.0.0 to 127.0.0.0/8 (255.0.0.0)
A (range in binary)	**00000000** .00000000.00000000.00000000 to**01111111**.00000000.00000000.00000000
A (mask in binary)	11111111.00000000.00000000.00000000/8
B (range/mask in dotted decimal)	**128** .0.0.0 to **191**.**255**.0.0/16 (255.255.0.0)
B (range in binary)	**10000000** .**00000000**.00000000.00000000 to**10111111**.**11111111**.00000000.00000000
B (mask in binary)	**11111111** .**11111111**.00000000.00000000/16
C (range/mask in dotted decimal)	**192** .**0**.**0**.0 to **223**.**255**.**255**.0/24 (255.255.255.0)
C (range in binary)	**11000000** .**00000000**.**00000000**.00000000 to**11011111**.**11111111**.**11111111**.00000000
C (mask in binary)	11111111.11111111.11111111.0000000/24
D[1] (range/mask in dotted decimal)	**224** .**0**.**0**.**0** to **239**.**255**.**255**.**255**/32 (255.255.255.255)

Class	Range
D (range in binary)	**11100000 .00000000.00000000.00000000** to**11101111.11111111.11111111.11111111**
D (mask in binary)	11111111.11111111.11111111.11111111/32
E[2] (range/mask in dotted decimal)	**240 .0.0.0** to **255.255.255.255**/32 (255.255.255.255)
E (range in binary)	**11110000 .00000000.00000000.00000000** to**11111111.11111111.11111111.11111111**
E (mask in binary)	11111111.11111111.11111111.11111111/32

Коли цифра, що потрапляє в маску мережі, змінюється від 1 до 0, або 0 до 1, мережна адреса змінюється. Наприклад, якщо змінити

10101100.00**0**10000.01011001.00100010 / 16 на

10101100.00**1**10000.01011001.00100010 / 16

ви змінили мережеву адресу з 172.16.89.34/16 до 172.48.89.34/16.

Коли цифра, яка виходить за межі мережевої маски, змінюється від 1 до 0, або 0 до 1, адреса хоста змінюється.

Наприклад, якщо ви зміните 10101100.00010000.01011001.00100010 / 16 на

10101100.00010000.01011001.00100011 / 16

ви змінили адресу хосту від 172.16.89.34/16 до 172.16.89.35/16.

Кожен клас IP-адреси підтримує певний діапазон IP-мережних адрес і IP-адрес.

Діапазон IP-адрес мережі, доступний для кожного класу, визначається за формулою - 2 * (кількість доступних бітів). У разі адрес класу A значення першого біта в 1-му октеті (як показано в Таблиця вище) фіксується на 0. Це залишає 7 бітів для створення додаткових мережевих адрес. Тому існує 128 IP-адреси мережі доступні для класу A ($2^7 = 128$).

Кількість адрес IP-вузлів, доступних для класу IP-адрес, визначається за формулою 2 * (кількість доступних бітів) - 2. У адресах класу A доступні 24 біти для адрес IP-хостів[2].

Тому для класу A доступні адреси 16,777,214 IP-хостів (2^{24}) - 2 = 16,777,214).

2 віднімається, оскільки існує 2 IP-адреси, які не можна використовувати для хосту. Адреса хоста всі 0 не можна використовувати, оскільки це те ж саме, що і мережева адреса. Наприклад, 10.0.0.0 не може бути IP-адресою хосту, оскільки є IP-адресою мережі. Адреса всі 1 - це широкомовна адреса, яка використовується для доступу до всіх хостів у мережі. Наприклад, IP-дейтаграма адресована 10.255.255.255 буде прийнята кожним хостом по мережі 10.0.0.0.

Довільний розподіл мережних і хост-бітів в класах IP-адрес призвело до неефективного розподілу простору IP. Наприклад, якщо ваша мережа має 16 окремих фізичних сегментів, вам знадобиться 16 IP адрес. Якщо ви використовуєте 16 мережевих адрес класу B, ви зможете підтримувати 65534 хостів на кожному з фізичних сегментів. Ваша загальна кількість підтримуваних IP адрес 1,048,544 (16 * 65,534 = 1,048,544).

Дуже мало мережевих технологій може масштабуватися до 65534 хостів на одному сегменті мережі. Дуже малій кількості компаній потрібно 1,048,544 IP-адрес. Ця проблема потребувала розробки нової стратегії. Дозволяється розподіл адрес IP-мереж на менші групи адрес IP-підмережі. Ця стратегія відома як підмережа.

Якщо у мережі є 16 окремих фізичних сегментів, знадобиться 16 адрес IP-підмереж. Це може бути виконано за допомогою однієї IP-адреси класу B. Наприклад, з IP-адреси класу B 172.16.0.0 може резервувати 4 біти з третього октету як біти підмережі. Це дає 16 IP-адрес підмережі $2^4 = 16$. У таблиці 2.2 наведено приклад адрес цих підмереж[1]. Перший стовпчик – номер підмережі, другий – її номер в десятинному вигляді, третій – бінарний номер підмережі.

Таблиця 2.2 – Підмережі для IP: 172.16.0.0/20

Number	IP Subnet Addresses in Dotted Decimal	IP Subnet Addresses in Binary
0[5]	172.16.0.0	10101100.00010000.00000000.00000000
1	172.16.16.0	10101100.00010000.00010000.00000000
2	172.16.32.0	10101100.00010000.00100000.00000000
3	172.16.48.0	10101100.00010000.00110000.00000000
4	172.16.64.0	10101100.00010000.01000000.00000000
5	172.16.80.0	10101100.00010000.01010000.00000000
6	172.16.96.0	10101100.00010000.01100000.00000000
7	172.16.112.0	10101100.00010000.01110000.00000000

Number	IP Subnet Addresses in Dotted Decimal	IP Subnet Addresses in Binary
8	172.16.128.0	10101100.00010000.10000000.00000000
9	172.16.144.0	10101100.00010000.10010000.00000000
10	172.16.160.0	10101100.00010000.10100000.00000000
11	172.16.176.0	10101100.00010000.10110000.00000000
12	172.16.192.0	10101100.00010000.11000000.00000000
13	172.16.208.0	10101100.00010000.11010000.00000000
14	172.16.224.0	10101100.00010000.11100000.00000000
15	172.16.240.0	10101100.00010000.11110000.00000000

У самому загальному сенсі термін «мережева адреса» означає «IP-адресу, яку маршрутизатори використовують для маршрутизації трафіку до певного сегмента мережі, щоб хост з призначеною IP-адресою в цьому сегменті міг його отримати».[2]

Тому термін адреса мережі може застосовуватися як до мережних, так і до підмережних адрес IP-мережі.

MAC-адреса (адреса обладнання) - це глобальна унікальна адреса, яка представляє мережеву карту і не може бути змінена. Адреса IPv4 відноситься до логічної адреси, яка є конфігурованою адресою, що використовується для ідентифікації мережі, до якої належить цей хост, а також до конкретного номера мережі. Іншими словами, адреса IPv4 складається з двох частин: мережева частина і хост-частина.

Це можна порівняти з вашою домашньою адресою. Лист, адресований Вашій домашній адресі, буде доставлений до Вашого будинку через цю логічну адресу. Якщо ви переїдете в інший будинок, ваша адреса зміниться, а листи, адресовані вам, будуть надіслані на вашу нову адресу. Але особа, якій приноситься лист, тобто «ви», залишається тим самим.

IPv4 адреси зберігаються внутрішньо як двійкові числа, але вони представлені в десяткових числах через простоту.

Багато адрес IPv4 зарезервовані, і ми не можемо використовувати ці адреси IPv4. Існує п'ять класів адрес IPv4 і певні спеціальні адреси.

2.2.1 Адреса IPv4 класу A

IPv4 адреси класу A призначені для дуже великих мереж. Перший октет IPv4-адреси "Клас A" використовується для ідентифікації мережі, і три залишкові октети використовуються для ідентифікації хоста в цій конкретній мережі (Network.Host.Host.Host).

32 біта IPv4-адреси "класу A" можуть бути представлені як 0xxxxxx.xxxxxxxx.xxxxxxx.xxxxxxxx.

Мінімально можливим значенням для лівого октету в двійкових файлах є 00000000 (десятковий еквівалент - 0), а максимально можливе значення для лівого октету - 01111111 (десятковий еквівалент - 127). Тому для адреси "Клас A" IPv4 крайній лівий октет повинен мати значення між 0-127 (0.X.X.X до 127.X.X.X).

Мережа 127.0.0.0 називається петлевою мережею. Адреса IPv4 127.0.0.1 використовується головним комп'ютером для передачі

повідомлення назад. Зазвичай використовується для виправлення неполадок і тестування мережі.

Комп'ютери, які не підключені безпосередньо до Інтернету, не мають глобально унікальних адрес IPv4. Вони потребують адрес IPv4, унікальних лише для цієї мережі. Мережа 10.0.0.0 належить "класу А", зарезервована для приватного використання і може використовуватися всередині будь-якої організації.

2.2.2 Адреса IPv4 класу B

"Клас B" IPv4 адреси використовуються для мереж середнього розміру. Перші два октету адреси "Клас B" IPv4 використовуються для ідентифікації мережі, а інші два октету використовуються для ідентифікації хоста в цій конкретній мережі (Network.Network.Host.Host).

32 біта IPv4-адреси "класу B" можуть бути представлені як 10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.

Мінімально можливим значенням для лівого октету в двійкових файлах є 10000000 (десятковий еквівалент - 128), а максимально можливе значення для лівого октету - 10111111 (десятковий еквівалент - 191). Тому для адреси "Клас B" IPv4 крайній лівий октет має мати значення між 128-191 (128.X.X.X до 191.X.X.X).

Мережа 169.254.0.0 відома як APIPA (автоматична приватна адреса IPv4). Діапазон адрес IPIP4 APIPA використовується, коли клієнт налаштований на автоматичне отримання адреси IPv4 від сервера DHCP, який не зміг зв'язатися з сервером DHCP для динамічної адреси IPv4.

Мережі від 172.16.0.0 до 172.31.0.0 зарезервовані для приватного використання.

2.2.3 Адреса IPv4 класу C

IPv4-адреси "класу C" зазвичай використовуються для малих і середніх підприємств. Перші три октету IPv4-адреси "Клас C"

використовуються для ідентифікації мережі, а решта - один октет використовується для ідентифікації хоста в цій конкретній мережі (Network.Network.Networkt.Host).

32 біта IPv4-адреси класу C можуть бути представлені як 110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.

Мінімально можливим значенням для лівого октету в двійкових файлах є 11000000 (десятковий еквівалент - 192), а максимально можливе значення для лівого октету - 11011111 (десятковий еквівалент - 223). Тому для IPv4-адреси класу "C" крайній лівий октет повинен мати значення між 192-223 (192.X.X.X до 223.X.X.X).

Мережі, починаючи з 192.168.0.0 до 192.168.255.0, зарезервовані для приватного використання.

2.2.4 Адреса IPv4 класу D

Адреси IPv4 класу D відомі як адреси багатоадресної передачі IPv4. Multicasting - це методика, розроблена для передачі пакетів від одного пристрою до багатьох інших пристроїв, без будь-якого непотрібного дублювання пакетів. При багатоадресній передачі один пакет надсилається з джерела і реплікується за необхідністю в мережі для досягнення максимальної кількості кінцевих користувачів. Ви не можете призначити ці IPv4-адреси пристроям.

Перші чотири біти лівого октету мережі "Клас D" зарезервовано як "1110". Інші 28 бітів використовуються для ідентифікації групи комп'ютерів, для яких призначено багатоадресне повідомлення.

Мінімально можливим значенням для останнього більшого октету в двійкових файлах є 11100000 (десятковий еквівалент 224), а максимально можливе значення для лівого октету - 11101111 (десятковий еквівалент - 239). Тому для адреси "Клас D" IPv4 крайній лівий октет повинен мати значення між 224-239 (224.X.X.X до 239.X.X.X).

2.2.5 Адреса IPv4 класу E

Клас E використовується лише в експериментальних цілях, і ви не можете призначити ці адреси IPv4 вашим пристроям.

Перші чотири біти лівого октету мережі "Class E" зарезервовано як "1111".

Мінімально можливим значенням для останнього більшого октету в двійкових файлах є 11110000 (десятковий еквівалент 240), а максимально можливе значення для лівого октету - 11111111 (десятковий еквівалент 255). Тому для адреси "Клас E" IPv4 крайній лівий октет має мати значення між 240-255 (240.X.X.X до 255.X.X.X)[2].

2.3 Моніторинг мережі з використанням Python

Моніторинг мережі - це процес, який відстежує вашу внутрішню IT-інфраструктуру на потенційні проблеми. Система здатна виявити велику кількість конкретних проблем, які можуть вплинути на загальну продуктивність вашої мережевої інфраструктури. Коли виявляються будь-які проблеми, система моніторингу мережі буде попереджати вашого системного адміністратора або організації IT-служб, а також надавати додаткові інструменти для виправлення помилок, перш ніж вони стануть справжньою проблемою.[1]

Однак мережі не є статичними. Далеко від цього вони, мабуть, є однією з найбільш вільних частин всієї інфраструктури. За визначенням, мережа з'єднує різні частини разом, постійно передаючи трафік туди і назад. Є багато можливих причин, які можуть викликати зупинку мережі, такі як: припинення роботи апаратного забезпечення, помилки програмного забезпечення, помилки людини, незважаючи на їх найкращі наміри, і так далі. Потрібні способи, щоб переконатися, що мережа працює як очікувалося і повідомити, коли з мережею не все гаразд. Цим займаються системи моніторингу мережі, які можуть буду різної архітектури й способу

використання.

Доцільність таких системи для моніторингу, як правило, підтверджується шляхом раннього виявлення та повідомлення про помилки, несправності та перевищення порогових значень, і таким чином дозволяє негайне втручання. Крім того, IT-співробітникам більше не потрібно постійно стежити за всіма компонентами мережі, включаючи сервери, настільні комп'ютери, програми, трафік тощо. Таким чином, система моніторингу зберігає цінний час, який адміністратор може ефективно використовувати для інших завдань.

Системи для моніторингу також значно сприяють безпеці мережі. Якщо показники системи раптово істотно відрізняються від норми, це може бути важливим натяком для IT-персоналу про якусь спробу шкідливих або фішингових атак. Програмне забезпечення моніторингу мережі може бути легко інтегровано в існуючі концепції безпеки, які мають вірусні сканери, брандмауери тощо, щоб забезпечить додаткову безпеку.[1]

2.3.1 Моніторинг з використанням сокетів(sockets)

Програмування сокета - це спосіб підключення двох вузлів до мережі для взаємодії один з одним. Один сокет (вузол) слухає конкретний порт на IP, а інший - до іншого, щоб сформувати з'єднання. Сервер формує сокет-слухач, коли клієнт доходить до сервера. Вони являють собою реальні основи для перегляду веб-сторінок. У простіших термінах це сервер і клієнт.

Програмування сокета починається шляхом імпортування бібліотеки сокетів і створення простого сокета.

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('0.0.0.0', 9995))
```

socket.AF_INET – необхідний для отримання пакетів з сегменту мережі IPv4;

socket.SOCK_DGRAM – ми вказуємо, що отримуватимемо лише UDP дейтаграмми

```
while True:
        buf, addr = sock.recvfrom(1500)
```

Цим циклом, ми зможемо отримувати дані з порту, поки не завершимо програму.

```
for i in range(0, count):
    try:
        base = SIZE_OF_HEADER+(i*SIZE_OF_RECORD)
        data = struct.unpack('!IIIIHH',buf[base+16:base+36])
        input_int, output_int = struct.unpack('!HH', buf[base+12:base+16])
        nfdata[i] = {}
        nfdata[i]['saddr'] = inet_ntoa(buf[base+0:base+4])
        nfdata[i]['daddr'] = inet_ntoa(buf[base+4:base+8])
        nfdata[i]['pcount'] = data[0]
        nfdata[i]['bcount'] = data[1]
...
```

Наступний цикл, можна занести у відповідний клас, чи функцію, оскільки він необхідний для того, щоб розпакувати набір байт, що надійшли до нашого порту і перетворити їх на відповідну UDP дейтаграмму.

2.3.2 Моніторинг з використанням SNMP

Простий протокол керування мережею (SNMP) - це протокол шару додатків, визначений Радою архітектури Інтернету (IAB) в RFC1157 для обміну інформацією, управління між мережевими пристроями. Він є частиною набору протоколів (TCP/IP).

SNMP є одним з широко прийнятих протоколів для управління та моніторингу елементів мережі. Більшість мережевих елементів професійного класу постачаються з вбудованим агентом SNMP. Ці агенти повинні бути включені і налаштовані для зв'язку з системою управління мережею (NMS).[1]

Основні компоненти SNMP та їхні функціональні можливості

SNMP складається з:

- менеджера SNMP;
- керованих пристроїв;

- SNMP агента;
- база даних управлінської інформації в іншому випадку називається базою інформації управління (MIB).

Проаналізуємо основні функції кожного компоненту такої системи й з’ясуємо їх структуру та необхідність.

Менеджер SNMP:

Менеджер або система управління є окремим об'єктом, який відповідає за зв'язок з агентом SNMP, реалізованим мережевими пристроями. Зазвичай це комп'ютер, який використовується для запуску однієї або декількох систем управління мережею.

Основні функції менеджера SNMP:

- агенти запитів;
- отримує відповіді від агентів;
- встановлює змінні в агентах;
- визначає асинхронні події від агентів.

Керовані пристрої:

Керований пристрій або мережевий елемент є частиною мережі, що вимагає певної форми моніторингу та управління, наприклад, маршрутизатори, комутатори, сервери, робочі станції, принтери, UPS, тощо.

Агент SNMP:

Агент - це програма, яка упакована в мережевий елемент. Увімкнення агента дозволяє збирати базу даних керування інформацією з локального пристрою і надає їй доступ до менеджера SNMP, коли він запитується. Ці агенти можуть бути стандартними (наприклад, Net-SNMP) або специфічними для постачальника (наприклад, агентом HP insight).

Ключові функції агента SNMP:

- збирає управлінську інформацію про своє місцеве середовище;
- зберігає та отримує управлінську інформацію, як визначено в MIB;

- повідомляє про подію менеджеру;
- діє як проксі для деяких мережевих вузлів, які не підлягають SNMP.

База даних управлінської інформації або інформаційна база управління (MIB):

Кожен агент SNMP підтримує інформаційну базу даних, що описує параметри керованих пристроїв. Менеджер SNMP використовує цю базу даних для запиту агента на конкретну інформацію і додатково переводить інформацію, необхідну для системи управління мережею (NMS). Ця спільно використовувана база даних між агентом і менеджером називається інформаційною базою управління (MIB).

Як правило, ці MIB містять стандартний набір статистичних і контрольних значень, визначених для апаратних вузлів мережі. SNMP також дозволяє розширення цих стандартних значень з значеннями, специфічними для конкретного агента, за допомогою використання приватних MIB.

Коротше кажучи, файли MIB - це набір питань, які менеджер SNMP може запитати у агента. Агент збирає ці дані локально і зберігає їх, як визначено в MIB. Таким чином, менеджер SNMP повинен знати про ці стандартні та приватні питання для кожного типу агента.

3. АНАЛІЗ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

В бакалаврському проекті розроблена система для моніторингу стану станції за її IP адресою. Ця система зручна у використанні, оскільки має інтуїтивно-зрозумілий web-інтерфейс. Її можна розбити на дві частини:

- web-ресурс;
- моніторингова програма.

Одна частина спілкується з користувачем, а інша виконує його запит та повертає інформацію для подальшого аналізу й пошуку рішень вирішення проблем.

3.1 Створення web-ресурсу за допомогою Python Django

Розроблений web-ресурс складається з двох підсистем:

- клієнтської частини, яка необхідна для зручного і зрозумілого користування навіть людиною не знайомою з програмуванням;
- серверною частиною, яка отримує й опрацьовує усі запити користувача, узгоджує дії з програмою моніторингу й зберігає необхідні дані в базу даних, для подальшого їх використання.

3.1.1 Клієнтська частина системи

Клієнтська частина представлена у вигляді зручного сайту, на якому для отримання повного функціоналу необхідно зареєструватись. Для реєстрації необхідно мати електрону-пошту, задати собі Login та Password. Після чого необхідно знову ввести псевдонім та пароль, але вже у вікно входу на ресурс. Пройшовши цей етап, користувач матиме змогу не тільки дивитись скорочені записи минулих запитів, а вже перевірити доступ до будь-якої IP адреси.

Після переходу на сторінку створення нового запиту, необхідно у відповідне поле записати IP адресу або domain ім’я ресурсу, який потрібно

перевірити.

Реалізована частина необхідна для відображення всіх отриманих даних й подальшого їх аналізу, створення нових даних та нових аккаунтів для класифікування запитів за користувачами, що їх створили. У разі зникнення необхідності в даних вже реалізована функція їх видалення.

Структурна реалізація клієнтської частини рисунок 3.1.

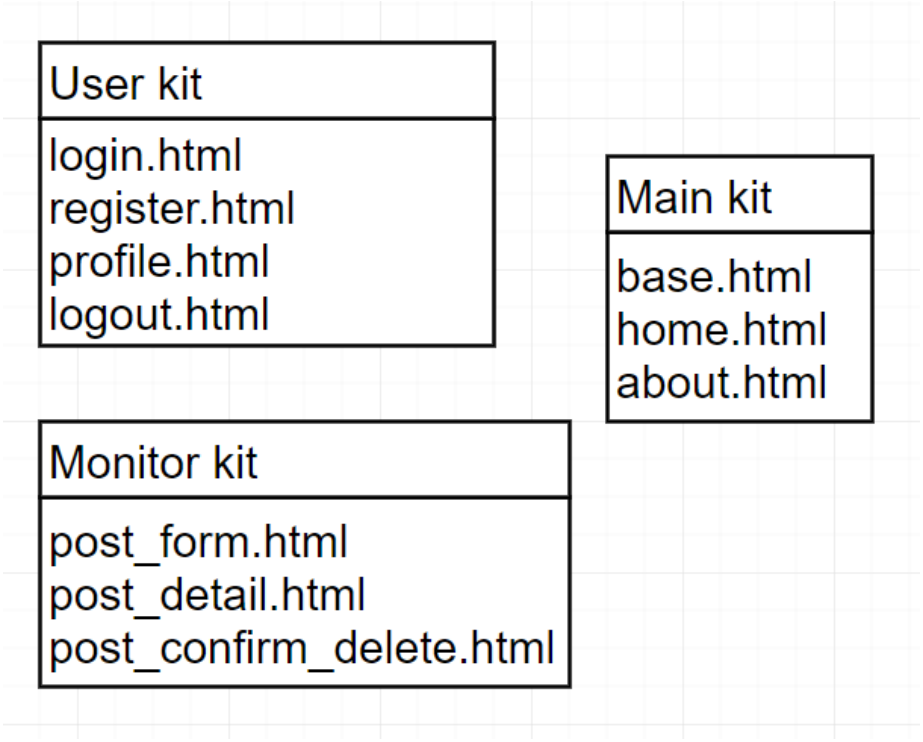

Рисунок 3.1 - Структурна реалізація клієнтської частини

Розглянемо кожен компонент клієнтської частини, його функції та представлення.

1. User kit – набір web-сторінок, призначених для реєстрації нових користувачів, або якщо аккаунт все створено, то використання саме його.

1.1 login.html – web сторінка, що містить форму для введення даних необхідних для авторизації користувача (рисунок 3.2).

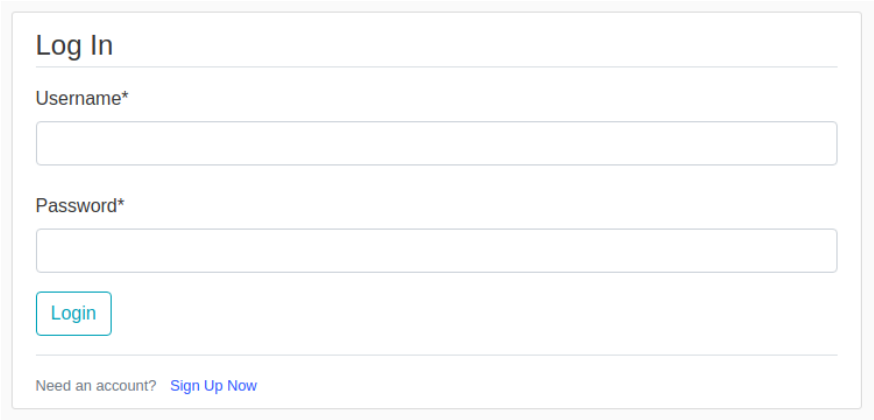

Рисунок 3.2 – Форма для авторизації на ресурс.

1.2 register.html – web сторінка, з формою для реєстрації користувача, після якої він отримає доступ до всіх ресурсів системи й матиме змогу моніторити IP адреси, які йому заманеться (рисунок 3.3).

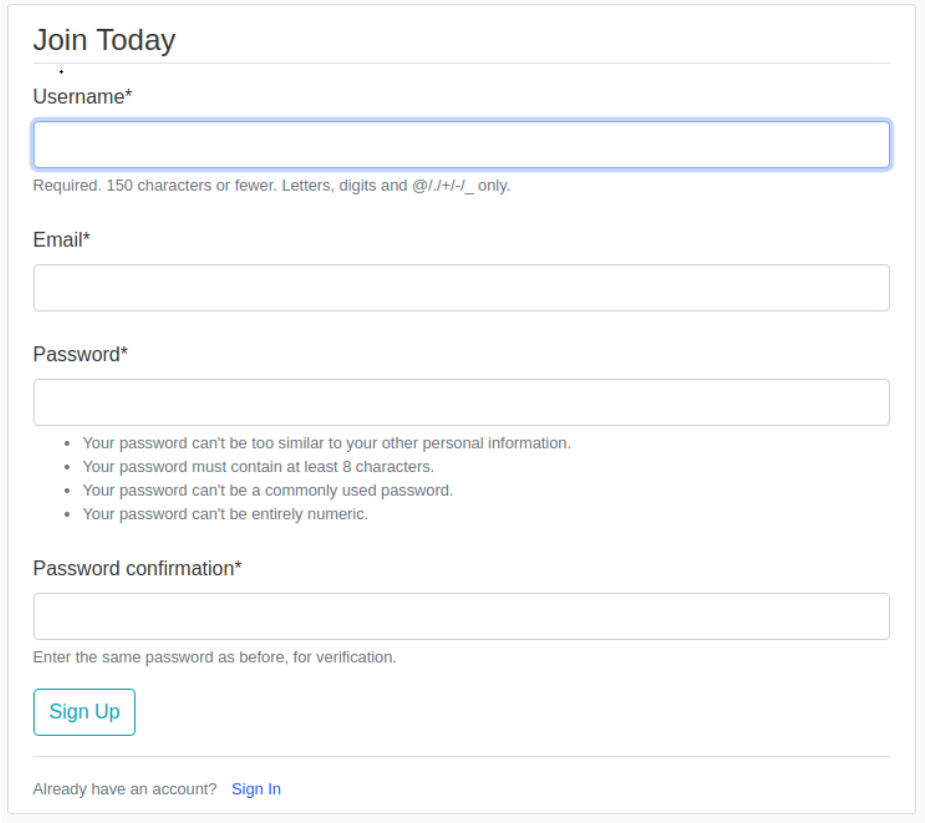

Рисунок 3.3 – Форма для реєстрації на ресурсі.

1.3 profile.html – web сторінка для зміни даних введених користувачем, у разі необхідності (рисунок 3.4).

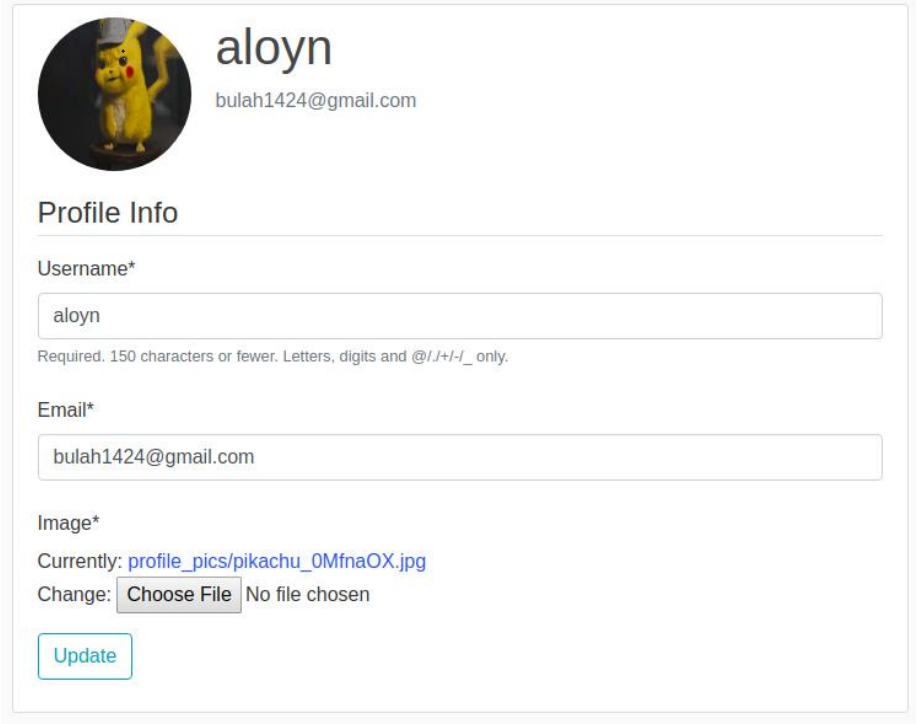

Рисунок 3.4 – Форма для зміни інформації про користувача.

1.4 logout.html – web сторінка для завершення сеансу роботи з моніторингом комп'ютерної мережі.

2 Monitor kit – набір web сторінок, призначених для роботи з моніторингом комп'ютерної мережі.

2.1 post_form.html – форма для створення нового запиту для моніторингу IP адреси, що передає дані на сервер (рисунок 3.5).

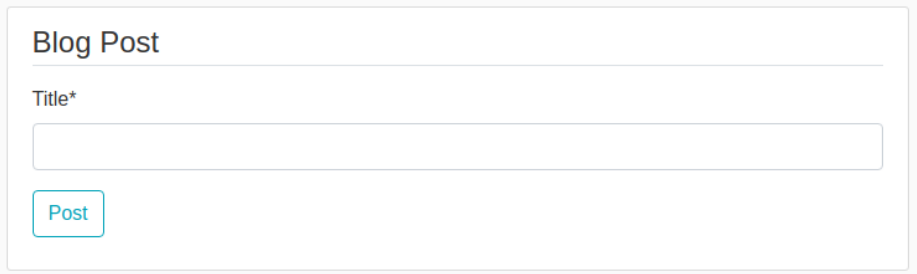

Рисунок 3.5 – Форма для початку моніторингу якоїсь IP адреси.

2.2 post_detail.html – web сторінка, що відображає дані, повернені сервером, для подальшого їх аналізу та зберігання (рисунок 3.6).

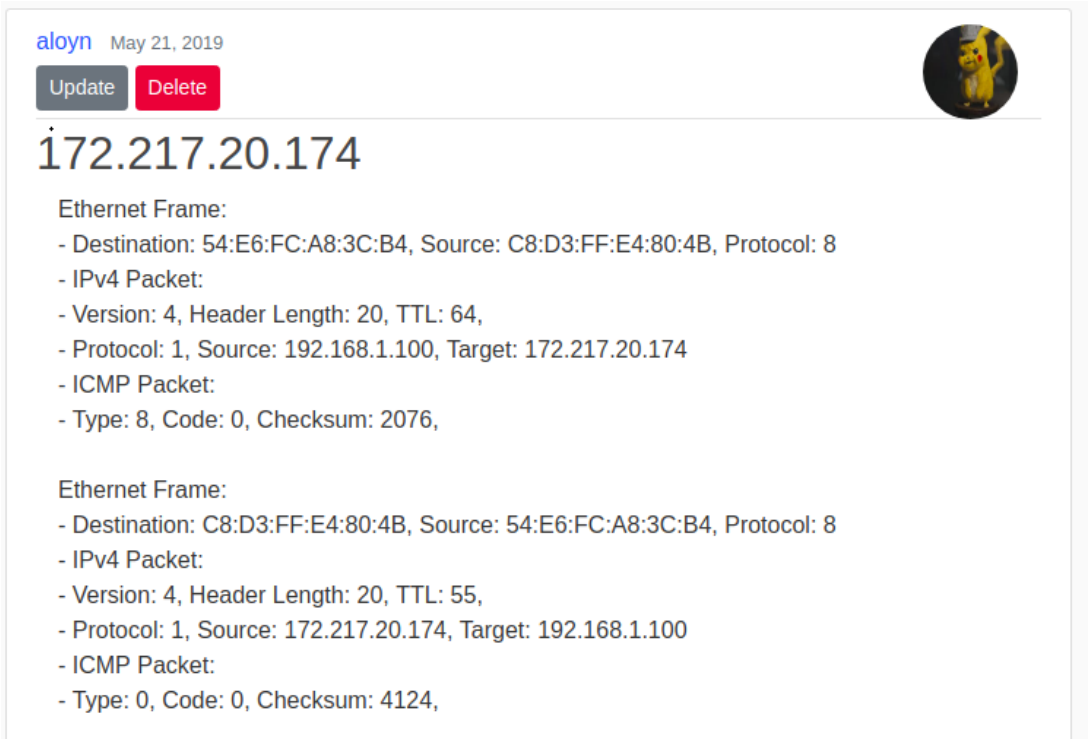

Рисунок 3.6 – Результат успішного моніторингу адреси.

2.3 post_confirm_delete.html – web сторінка, остаточного видалення інформації про будь-який запис створений авторизованим користувачем, якщо він виявився непотрібним.

3 Main kit – набір сторінок для зручного користування даним ресурсом.

3.1 base.html – необхідна для зручного переході між основними сторінками й містить посилання на сторінки користувача та на сторінки моніторингу.

3.2 home.html – web сторінка на якій відображенні всі попередні запити, але в скороченому форматі, оскільки повну інформацію може мати лише зареєстрований користувач, а ця сторінка відкривається при переході на наш ресурс і доступна для кожного (рисунок 3.7).

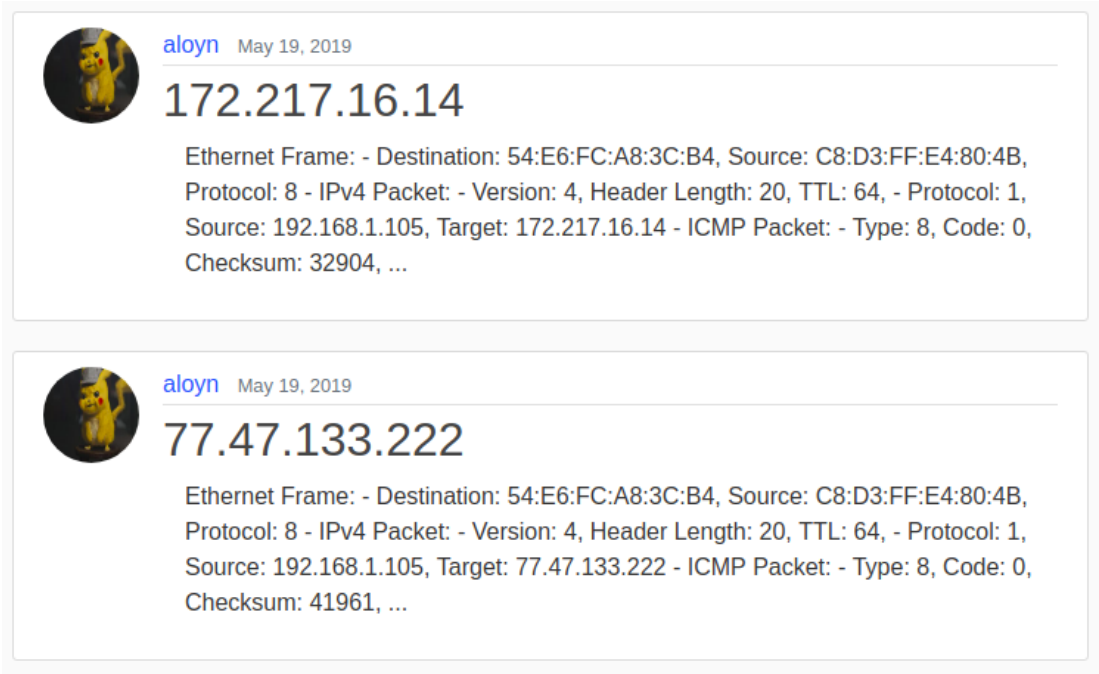

Рисунок 3.7 – Доступна інформація незареєстрованому користувачеві.

3.3 about.html – сторінка на якій адміністратор може розмістити важливу інформацію для користувачів.

3.1.2 Серверна частина системи

Серверна частина представлена у вигляді декількох контролерів, реалізованих за допомогою фреймворку Django. Вони призначені для обробки запитів, що прийшли з клієнтської частини системи. Усі дані оброблені серверною частиною зберігаються у базі даних, а даному проекті була використана SQlite. У разі введення неправильних даних користувачем, сервер під час обробки запиту, знайде їх та поверне повідомлення з помилкою, яку було допущено.

Структура серверної частини системи на рисунку 3.8.

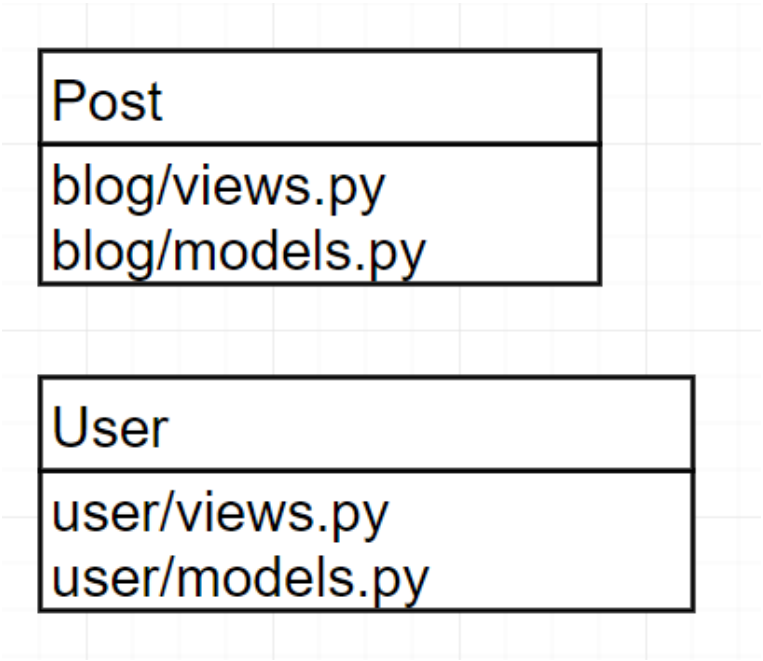

Рисунок 3.8 – Серверна частина системи.

1. Post – набір методів для обробки інформації переданої користувачем, про адресу, яку необхідно перевірити.

1.1 blog/models.py – модель полями, створена для відображення, обробки та збереження у базі даних інформації про моніториг, який був запитаний користувачем.

1.2 blog/views.py – набір методів та класів для перевірки правильності введених даних, передання їх на обробку програмі моніторигу, збереження в базі даних та виведення на клієнтський рівень результату роботі програми.

Нижче наведено метод для створення нового запиту моніторингу IP адрес рисунок 3.9.

```python
class PostCreateView(LoginRequiredMixin, CreateView):
        model = Post
        fields = ['title']

        def form_valid(self, form):
                form.instance.user = self.request.user
                subprocess.run(['python', 'Sniffer/sniffer.py', form.instance.title], shell=False)
                f_o = open('./Sniffer/some_data.txt', 'r')
                form.instance.message = f_o.read()
                if form.instance.message == '':
                        form.instance.message = 'Server unavailable!!!'
                f_o.close()
                if not validate_ip(form.instance.title):
                        f_o = open('./Sniffer/ip.txt', 'r')
                        form.instance.title += ' (' + f_o.read().rstrip() + ')'
                        f_o.close()
                return super().form_valid(form)
```

Рисунок 3.9 – Метод для створення нового запиту.

Метод який визначає, що ми отримали IP чи Domain ім’я:

```python
def validate_ip(s):
        a = s.split('.')
        if len(a) != 4:
                return False
        for x in a:
                if not x.isdigit():
                        return False
                i = int(x)
                if i < 0 or i > 255:
                        return False
        return True
```

Рисунок 3.10 – Перевірка IP.

2 User – набір методів для створення користувача, перегляду детальної інформації та її редагування у разі необхідності.

2.1 user/model.py – модель користувача, яка використовується при створенні, редагуванні та видалення користувача з бази даних, до якої можна додати будь-які поля якщо вони знадобляться (рисунок 3.11).

```
class Profile(models.Model):
        user = models.OneToOneField(User, on_delete=models.CASCADE)
        image = models.ImageField(default='default.jpg', upload_to='profile_pics')

        def __str__(self):
                return f'{self.user.username} Profile'

        def save(self):
                super().save()

                img = Image.open(self.image.path)

                if img.height > 300 or img.width > 300:
                        output_size = (300, 300)
                        img.thumbnail(output_size)
                        img.save(self.image.path)
```

Рисунок 3.11 – Модель користувача.

2.2 user/views.py – набір методів та класів для створення нового користувача, або зміни даних вже створеного та збереження цього. При введені некоректних даних на клієнтський рівень буде передане повідомлення з помилкою допущенною користувачем (рисунок 3.12).

```
def register(request):
        if request.method == 'POST':
                form = UserRegisterForm(request.POST)
                if form.is_valid():
                        form.save()
                        username = form.cleaned_data.get('username')
                        messages.success(request, f'Your account has been
                        return redirect('login')
        else:
                form = UserRegisterForm()
        return render(request, 'users/register.html', {'form': form})
```

Рисунок 3.12 – Створення нового користувача.

3.2 Створення програми моніторингу стану мережі

Моніторингова програма представлена у вигляді набору методів та класів для перевірки стану необхідної адреси переданої з рівня серверної частини.

У разі коли програма отримала адресу у вигляді IP, вона відразу починає формувати ICMP повідомлення та відправляти їх за цією адресою. Але якщо замість цього ми отримали domain ім’я, то вона спочатку перетворює її на IP адресу, за допомогою DNS серверів, які на необхідний запит з цим ім’ям і повернуть нам його IP.

Після цього починається моніторинг усіх прийнятих й відправлених пакетів нашим вузлом й пошук тих, що мають адресу відправника, або отримувача рівною той, що була передана нам. Знайдені пакети запам’ятовуються у повному своєму вигляді й передаються на серверний рівень, а далі й на клієнтський рівень, де користувач, зможе їх проаналізувати.

Клас для відображення та формування ICMP повідомлень наведений на рисунок 3.13.

```
class ICMP:

    def __init__(self, raw_data):
        self.type, self.code, self.checksum = struct.unpack('! B B H', raw_data[:4])
        self.data = raw_data[4:]
```

Рисунок 3.13 – Клас ICMP повідомлень.

3.3 Опис роботи і зв’язків ресурсу та програми моніторингу

Розроблений програмний комплекс є системою, що складається з трьох модулів:

1) клієнтська частина системи;
2) серверна частина системи;
3) моніторингова програма;

Структура цього комплексу зображена на рисунку. 3.14.

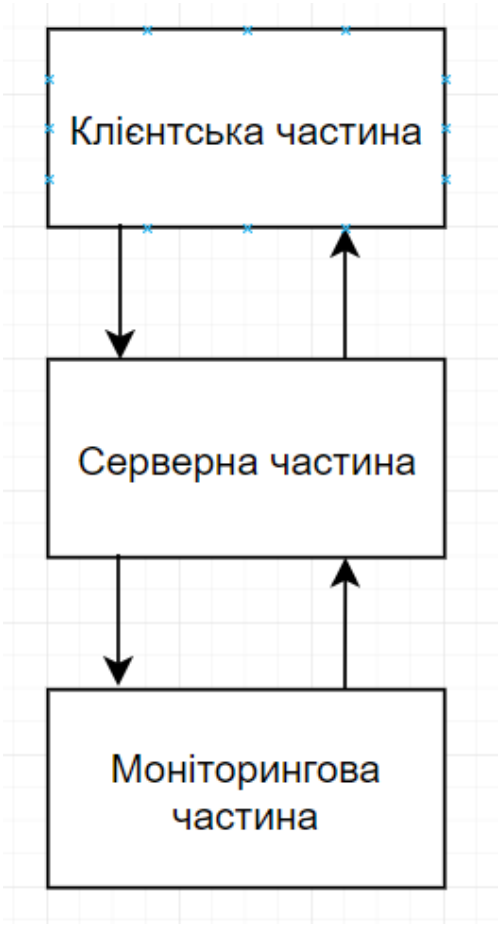

Рисунок 3.14 Загальна структура системи.

Сумісна робота клієнт-серверної системи на Django та програми моніторингу мережі, можлива завдяки функції subprocess(). Вона запускає нашу програму, якщо клієнт заповнив форму й відправив її на серверну частину. На серверній частині запит формується й виконується виклик цієї функції з необхідними параметрами. Після чого на іншому потоці запускається наша програма, яка перевіряє стан адреси й повертає дані на серверну частину. Вона їх зберігає у базі даних, після чого передає на відображення клієнтському рівню. Після всіх цих дій користувач, може подивитись як відбувалось «листування» між сервером та адресою, яку він ввів. Та проаналізувати час затримки.

3.4. Приклад роботи. Демонстрування можливостей системи

Користувач з будь-якого пристрою, який має вихід до комп'ютерної мережі, заходить на мій web-ресурс. Проходить етап реєстрації, заповнивши

необхідну форму, клієнтська частина формує з цих даних повідомлення й відправляє серверній частині для перевірки правильності даних й створення аккаунти користувача. Після цього користувач перенаправляється на сторінку для входу на ресурс, за допомогою даних, які щойно заповнив у попередній формі. У разі успішного проходження даного етапу він матиме змогу користуватись повними можливостями системи.

Для початку моніторингу необхідно перейти на сторінку New Post та ввести IP адресу або domain ім’я ресурсу, який ви хочете перевірити. Дані, що були заповнені, передаються на серверний рівень, який їх запам’ятовує та передає на рівень програми моніторингу. Ця програма перевіряє дані, якщо прийшло domain ім’я, то вона перетворює його на IP адресу, за допомогою з’єднання з DNS серверами. Вона використовується, як адреса отримувача для формування ICMP повідомлень, які необхідні для встановлення зв’язку с ресурсом.

У разі не отримання відповіді, яка представляє собою ICMP повідомлення з вказаною користувачем адресою вже на місці адреси відправника, відправляємо на серверний рівень повідомлення про помилку у з’єднані, який у свою чергу повідомляє про це користувача.

Якщо ми отримали необхідну ICMP відповідь, ми запам’ятовуємо усі пакети, що нам відправив ресурс за адресою користувача й повертаємо їх у повному розмірі на серверний рівень, який запам’ятовує їх в базі даних та відправляє на клієнтський рівень.

Клієнтська частина системи відображає у зручному форматі ICMP повідомлення для їх аналізу, або помилку якщо сервер не доступний.

ВИСНОВКИ

В ході виконання даного дипломного проекту було розглянуто IP-мережу, а саме як вона влаштована, її основні компоненти та протоколи. Дізналися основні способи моніторингу та аналізу стану такої мережі та її недоліки. На основі цього було створено клієнт-серверну систему моніторингу та аналізу стану IP-мережі.

У першому розділі було вивчено теоретичне підґрунтя для розуміння складових комп'ютерної мережі, основних протоколів, які використовуються для передачі даних. Використовуючи ці знання було вирішено створити систему, яка перевірятиме стан з'єднання між вузлами IP-мережі за допомогою ICMP запитів й аналізуючи їх приймати якісь рішення.

Було проаналізовано декілька способів реалізації моніторингу пакетів серед яких використання допоміжного протоколу SNMP або socket-моніторинг. Був обраний саме другий варіант, оскільки він не потребував стороннього програмного забезпечення та додаткових апаратних пристроїв.

У другому розділі було обрано мову програмування, на котрій й буде описана всю програма моніторингу. А також обраний фреймворк, для написання клієнт-серверної системи, представленої у вигляді веб-ресурсу. Мовою став Python, а фреймворком Django.

У третьому розділі було продемонстровано створену програму, інструкцію, як нею користуватись, а також наведено декілька прикладів її роботи.

Ця система зручна для використання й може використовуватися не лише для закріплення знань курсу «Комп'ютерних мереж», а й у повсякденному житті, коли необхідно дізнатись причину відсутності з'єднання с якоюсь станцією сегменту IP мережі, що використовую протокол IPv4.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Eric Chou. Mastering Python Networking: book. Published by Packt Publishing Ltd. Livery Place, 35 Livery Street, Birmingham, 2017.
2. Cisco Systems, Inc. IP Addressing: IPv4 Addressing Configuration Guide, Cisco IOS XE Release 3S. 170 West Tasman Drive, San Jose, CA 95134-1706, USA, 2015.
3. Класифікація комп'ютерних мереж – [Електронний ресурс]. – Режим доступу: https://www.belden.com/blog/smart-building/11-types-of-networks-explained-vpn-lan-more
4. WAN та LAN мережі – [Електронний ресурс]. – Режим доступу: https://www.computernetworkingnotes.com/networking-tutorials/types-of-computer-network-explained-in-easy-language.html
5. Модель OSI та її рівні – [Електронний ресурс]. – Режим доступу: https://www.webopedia.com/quick_ref/OSI_Layers.asp